

# Algorithmes rapides pour l'évaluation polynomiale sur les nombres flottants

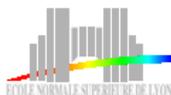
—

Guillaume Revy

Encadrants : Gilles Villard & Claude-Pierre Jeannerod

Projet Arénaire (LIP, ENS-Lyon)

22/01/2007



## Contexte

**Problème** : comment évaluer un polynôme le plus rapidement possible en un point quelconque, à une certaine précision ?

**Solutions classiques** : soit  $a(x) = a_3x^3 + a_2x^2 + a_1x + a_0$ . Quel schéma utiliser pour évaluer ce polynôme ?

▶ **Schéma de Horner** :  $((a_3x + a_2)x + a_1)x + a_0$

→  $n$  multiplications /  $n$  additions

→ méthode précise mais séquentielle

▶ **Schéma de Estrin** :  $(a_3x + a_2)x^2 + (a_1x + a_0)$

→  $n + \log(n + 1) - 1$  multiplications /  $n$  additions

→ plus de multiplications que Horner mais évaluation parallélisable

## Problématique

Il existe des méthodes nécessitant moins d'opérations :

⇒ reposent sur une reformulation du polynôme : **phase de préparation**

⇒ préparation : effectuée **une fois pour toutes** avant les évaluations

▶ **Schéma de Knuth & Eve** :  $\frac{n}{2}$  multiplications /  $n$  additions

▶ **Schéma de Paterson & Stockmeyer** :  $\frac{n}{2} + \log(n)$  multiplications /  $\frac{3n}{2}$  additions

⇒ méthodes à base de préparation : intéressante dans le cadre de l'évaluation des fonctions élémentaires ( $\log$ ,  $\exp$ ,  $\cos$ ,  $\dots$ )

→ on peut se permettre une phase de préparation, même coûteuse

## Extension aux schémas hybrides

- ▶ Soit un polynôme  $a(x) = \sum_{i=0}^n a_i x^i$  :

$$a(x) = \underbrace{\left( \sum_{i=k}^n a_i x^{i-k} \right)}_{\text{méthode rapide}} \cdot x^k + \sum_{i=0}^{k-1} a_i x^i$$

méthode précise

- combinaison méthode rapide et peu précise / méthode « lente » et précise
- ▶ Mise en évidence de l'efficacité des schémas hybrides
  - compenser les pertes de précision
  - bon compromis précision / rapidité

# Plan de l'exposé

Étude de l'algorithme de Knuth & Eve

Schémas hybrides et fonctions élémentaires

Conclusions et travaux en cours

## Phase de préparation dans le cas $n$ impair

**Entrées** :  $a_n, a_{n-1}, \dots, a_1, a_0$   
**Sorties** :  $\alpha_1, \dots, \alpha_m, \beta_0, \dots, \beta_m$

### 1. Découpage $a$ en une partie paire $g$ et une partie impaire $h$

$$a(x) = g(y) + x \cdot h(y), \text{ avec } y = x^2 \text{ et } m = \deg(h) = \frac{n-1}{2}.$$

### 2. Calcul des $m$ racines $\alpha_1, \dots, \alpha_m$ de $h$ :

$$a(x) = g(y) + a_n x \prod_{i=1}^m (y - \alpha_i).$$

### 3. Divisions successives par $y - \alpha_1, \dots, y - \alpha_m$ :

→ soit  $\beta_0$  le reste de la division de  $g$  par  $y - \alpha_1$  et  $g^{(1)}$  le quotient, alors :

$$a(x) = (g^{(1)}(y) + a_n x \prod_{i=2}^m (y - \alpha_i))(y - \alpha_1) + \beta_0.$$

⇒ en continuant le processus avec  $g^{(1)}$ , on obtient  $\beta_0, \beta_1, \beta_2, \dots, \beta_m$

## Phase de préparation dans le cas $n$ impair

**Entrées** :  $a_n, a_{n-1}, \dots, a_1, a_0$   
**Sorties** :  $\alpha_1, \dots, \alpha_m, \beta_0, \dots, \beta_m$

1. Découpage  $a$  en une partie paire  $g$  et une partie impaire  $h$

$$a(x) = g(y) + x \cdot h(y), \text{ avec } y = x^2 \text{ et } m = \deg(h) = \frac{n-1}{2}.$$

2. Calcul des  $m$  racines  $\alpha_1, \dots, \alpha_m$  de  $h$  :

$$a(x) = g(y) + a_n x \prod_{i=1}^m (y - \alpha_i).$$

3. Divisions successives par  $y - \alpha_1, \dots, y - \alpha_m$  :

→ soit  $\beta_0$  le reste de la division de  $g$  par  $y - \alpha_1$  et  $g^{(1)}$  le quotient, alors :

$$a(x) = (g^{(1)}(y) + a_n x \prod_{i=2}^m (y - \alpha_i))(y - \alpha_1) + \beta_0.$$

⇒ en continuant le processus avec  $g^{(1)}$ , on obtient  $\beta_0, \beta_1, \beta_2, \dots, \beta_m$

## Phase de préparation dans le cas $n$ impair

**Entrées** :  $a_n, a_{n-1}, \dots, a_1, a_0$   
**Sorties** :  $\alpha_1, \dots, \alpha_m, \beta_0, \dots, \beta_m$

1. Découpage  $a$  en une partie paire  $g$  et une partie impaire  $h$

$$a(x) = g(y) + x \cdot h(y), \text{ avec } y = x^2 \text{ et } m = \deg(h) = \frac{n-1}{2}.$$

2. Calcul des  $m$  racines  $\alpha_1, \dots, \alpha_m$  de  $h$  :

$$a(x) = g(y) + a_n x \prod_{i=1}^m (y - \alpha_i).$$

3. Divisions successives par  $y - \alpha_1, \dots, y - \alpha_m$  :

→ soit  $\beta_0$  le reste de la division de  $g$  par  $y - \alpha_1$  et  $g^{(1)}$  le quotient, alors :

$$a(x) = (g^{(1)}(y) + a_n x \prod_{i=2}^m (y - \alpha_i))(y - \alpha_1) + \beta_0.$$

⇒ en continuant le processus avec  $g^{(1)}$ , on obtient  $\beta_0, \beta_1, \beta_2, \dots, \beta_m$

## Phase de préparation dans le cas $n$ impair

**Entrées** :  $a_n, a_{n-1}, \dots, a_1, a_0$   
**Sorties** :  $\alpha_1, \dots, \alpha_m, \beta_0, \dots, \beta_m$

1. Découpage  $a$  en une partie paire  $g$  et une partie impaire  $h$

$$a(x) = g(y) + x \cdot h(y), \text{ avec } y = x^2 \text{ et } m = \deg(h) = \frac{n-1}{2}.$$

2. Calcul des  $m$  racines  $\alpha_1, \dots, \alpha_m$  de  $h$  :

$$a(x) = g(y) + a_n x \prod_{i=1}^m (y - \alpha_i).$$

3. Divisions successives par  $y - \alpha_1, \dots, y - \alpha_m$  :

→ soit  $\beta_0$  le reste de la division de  $g$  par  $y - \alpha_1$  et  $g^{(1)}$  le quotient, alors :

$$a(x) = (g^{(1)}(y) + a_n x \prod_{i=2}^m (y - \alpha_i))(y - \alpha_1) + \beta_0.$$

⇒ en continuant le processus avec  $g^{(1)}$ , on obtient  $\beta_0, \beta_1, \beta_2, \dots, \beta_m$

## Phase de préparation dans le cas $n$ impair

4. Finalement, avec  $y = x^2$  et  $\beta_m = a_n x + \beta_m^{(0)}$ , la nouvelle formule de  $a(x)$  est :

$$a(x) = (\dots ((\beta_m(y - \alpha_m) + \beta_{m-1})(y - \alpha_{m-1}) + \beta_{m-2}) \dots)(y - \alpha_1) + \beta_0$$

→ intérêt si les racines  $\alpha_i$  sont réelles

► Généralisation au cas  $n$  pair :  $\beta_m = (a_n x + \beta_m^{(1)})x + \beta_m^{(0)}$ .

## Phase de préparation dans le cas $n$ impair

4. Finalement, avec  $y = x^2$  et  $\beta_m = a_n x + \beta_m^{(0)}$ , la nouvelle formule de  $a(x)$  est :

$$a(x) = \left( \dots \left( (\beta_m(y - \alpha_m) + \beta_{m-1})(y - \alpha_{m-1}) + \beta_{m-2} \right) \dots \right) (y - \alpha_1) + \beta_0$$

→ intérêt si les racines  $\alpha_i$  sont réelles

- Généralisation au cas  $n$  pair :  $\beta_m = (a_n x + \beta_m^{(1)})x + \beta_m^{(0)}$ .

## Cas général ( $n \geq 3$ )

- ▶ De manière générale, la partie impaire  $h$  du polynôme  $a$  n'a pas que des racines réelles.

**Théorème [J. Eve, 1964]** : si on décale suffisamment le polynôme  $a$ , la partie impaire  $h$  du polynôme décalé n'a que des racines réelles.

- ▶ Phase de préparation

1. détermination d'un décalage  $c$  et calcul des coefficients de  $f(x) = a(x - c)$
2. application de l'algorithme de préparation au polynôme décalé  $f(x)$
3. évaluation de  $f$  en  $x + c$

⇒ deux types de décalages (Eve / Knuth)

⇒ décalage de Knuth  $\rightarrow \beta_0 = 0$

- ▶ Schéma général d'évaluation avec  $y = (x + c)^2$  :

$$a(x) = (\dots ((\beta_m(y - \alpha_m) + \beta_{m-1})(y - \alpha_{m-1}) + \beta_{m-2}) \dots)(y - \alpha_1) + \beta_0$$

## Cas général ( $n \geq 3$ )

- ▶ De manière générale, la partie impaire  $h$  du polynôme  $a$  n'a pas que des racines réelles.

**Théorème [J. Eve, 1964]** : si on décale suffisamment le polynôme  $a$ , la partie impaire  $h$  du polynôme décalé n'a que des racines réelles.

- ▶ Phase de préparation

1. détermination d'un décalage  $c$  et calcul des coefficients de  $f(x) = a(x - c)$
2. application de l'algorithme de préparation au polynôme décalé  $f(x)$
3. évaluation de  $f$  en  $x + c$

⇒ deux types de décalages (Eve / Knuth)

⇒ décalage de Knuth  $\rightarrow \beta_0 = 0$

- ▶ Schéma général d'évaluation avec  $y = (x + c)^2$  :

$$a(x) = (\dots ((\beta_m(y - \alpha_m) + \beta_{m-1})(y - \alpha_{m-1}) + \beta_{m-2}) \dots)(y - \alpha_1) + \beta_0$$

## Cas général ( $n \geq 3$ )

- ▶ De manière générale, la partie impaire  $h$  du polynôme  $a$  n'a pas que des racines réelles.

**Théorème [J. Eve, 1964]** : si on décale suffisamment le polynôme  $a$ , la partie impaire  $h$  du polynôme décalé n'a que des racines réelles.

- ▶ Phase de préparation

1. détermination d'un décalage  $c$  et calcul des coefficients de  $f(x) = a(x - c)$
2. application de l'algorithme de préparation au polynôme décalé  $f(x)$
3. évaluation de  $f$  en  $x + c$

⇒ deux types de décalages (Eve / Knuth)

⇒ décalage de Knuth  $\rightarrow \beta_0 = 0$

- ▶ Schéma général d'évaluation avec  $y = (x + c)^2$  :

$$a(x) = (\dots ((\beta_m(y - \alpha_m) + \beta_{m-1})(y - \alpha_{m-1}) + \beta_{m-2}) \dots)(y - \alpha_1) + \beta_0$$

## Analyse du schéma

- ▶ Complexité de l'évaluation :  $\lfloor \frac{n}{2} \rfloor + 2$  multiplications et  $n$  additions  
→ plus rapide que Horner dès que  $n \geq 5$
- ▶ Dans le *modèle flottant standard* : algorithme stable
  - ▶ *backward stable* : valeur calculée = solution exacte d'un problème proche
  - ▶ si les racines sont connues exactement  
→ erreur *backward* majorée par  $\gamma_{3n} \approx 3n \times 2^{-p}$  ( $p$  : précision)
  - ▶ si les racines sont connues avec arrondi correct  
→ erreur *backward* majorée par  $\gamma_{3n+1}$
  - ▶ si  $x + c$  est calculable exactement  
→ erreur *backward* majorée par  $\gamma_{2n}$  (comme Horner)

## Résultats numériques et observations

### Observation générale sur des polynômes aléatoires

- ▶ Malgré un décalage nul : erreur absolue pour Knuth & Eve plus élevée que pour Horner.

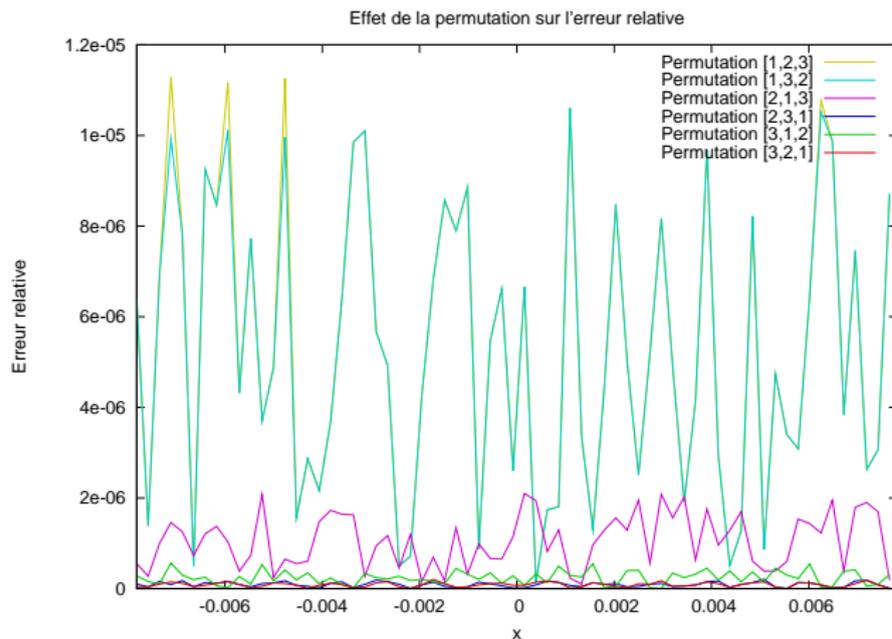
	$n = 3$	$n = 4$	$n = 5$	$n = 6$	$n = 7$	$n = 8$
Ratio des nombres de conditionnement	490	470	1400	1400	2200	1900
Ratio des erreurs absolues	75	65	254	261	357	344

TAB.: Ratio KE/Horner, pour l'évaluation de polynômes de degré  $n \in \{3, \dots, 8\}$  sur l'intervalle  $[-2^{-7}, 2^{-7}]$

### Effet de la permutation

- ▶ Importance de l'ordre du traitement des racines
  - effet sur la valeur des  $\beta_i$
  - variation de l'erreur relative en fonction de la permutation

## Résultats numériques et observations



Erreur relative max.  
([1,2,3])

$$\rightarrow e \approx 1.2 \times 10^{-5}$$

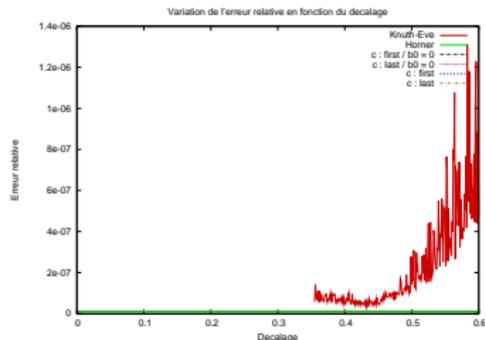
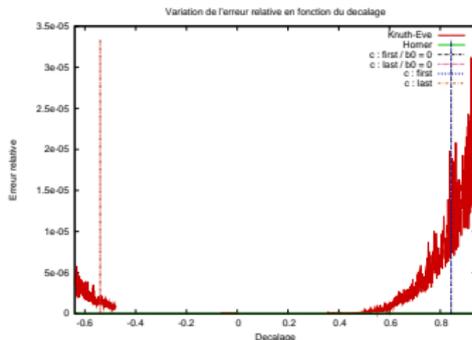
Erreur relative min.  
([3,2,1])

$$\rightarrow e \approx 2.0 \times 10^{-7}$$

## Résultats numériques et observations

### Effet du décalage

- ▶ De manière générale, plus le décalage est élevé, plus l'erreur relative est importante

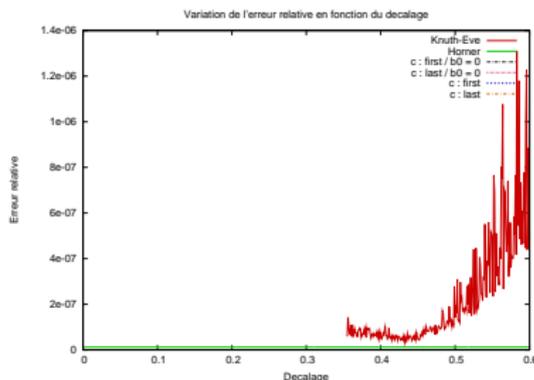


- ▶ Décalages de Eve et Knuth : non optimaux / pessimistes

## Résultats numériques et observations

### Effet du décalage

- ▶ Dans certains cas : effet positif du décalage (degré 7 /  $[-2^{-7}, 2^{-7}]$ )

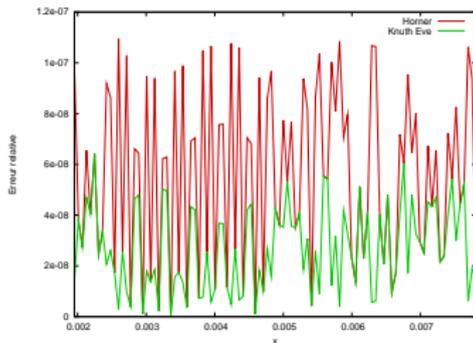


- ▶  $c = 0 \rightarrow e \approx 4.9 \times 10^{-8}$
- ▶  $c$  de Knuth/Eve  $\approx 8.4 \times 10^{-1} \rightarrow e \approx 9.8 \times 10^{-6}$
- ▶ meilleur  $c \approx 4.3 \times 10^{-1} \rightarrow e \approx 3.2 \times 10^{-8}$  (Horner :  $e \approx 3.1 \times 10^{-8}$ )

## Résultats numériques et observations

### Cas où le décalage est nul

- ▶ Analyse d'erreur :  $c = 0 \rightarrow$  même borne d'erreur *backward* que Horner
  - $\rightarrow$  dans certains cas, comparable à Horner
  - $\rightarrow$  dans certains cas, meilleur que Horner
- ▶ Évaluation d'un polynôme de degré 12
  - $\rightarrow [2^{-9}, 2^{-7}]$
  - $\rightarrow c = 0$

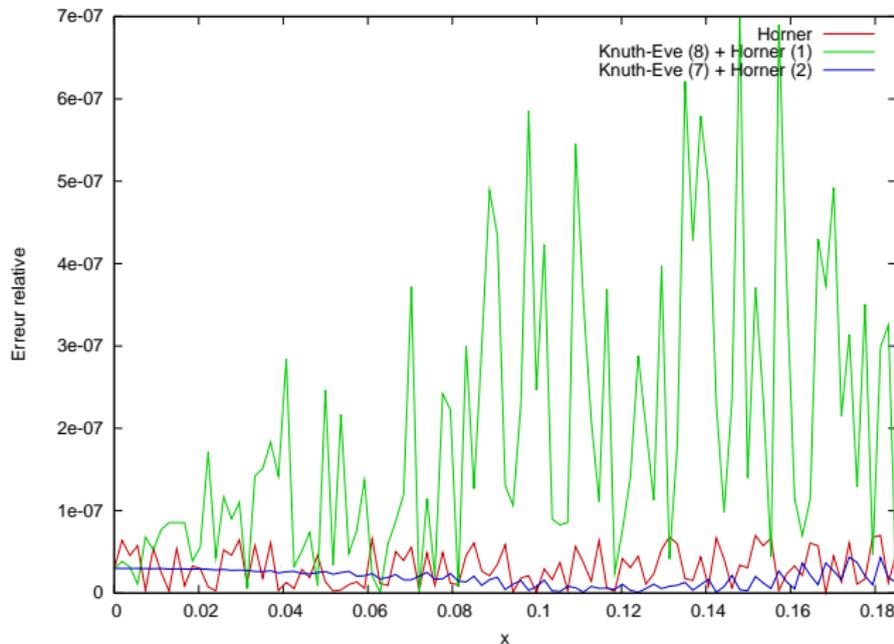


- ▶ évaluation par Knuth & Eve :  
 $e \approx 6.4 \times 10^{-8}$
- ▶ évaluation par Horner :  
 $e \approx 1.1 \times 10^{-7}$

## Conclusion sur l'algorithme de Knuth & Eve

- ▶ Comparaison par rapport à Horner
  - plus rapide que Horner (moins d'opérations) dès que  $n \geq 5$
  - dans certains cas : aussi précis que Horner
  
- ▶ Nombreuses réécritures possibles
  - difficulté de déterminer rapidement une meilleure réécriture

## Efficacité des schémas hybrides



## Exemple 1 – Accélération de la fonction $\arcsin(x)$

### Fonction $\arcsin(x)$ existante

- ▶ fonction approchée sur  $[0, 1]$  par 10 polynômes de degré 9, 14 ou 17
- ▶ tous évalués par un schéma de Horner

### Solutions proposées

- ▶ schéma hybride Knuth & Eve / Estrin pour le premier polynôme
- ▶ schéma de Estrin pour les autres polynômes

⇒ gain : 73 à 83 cycles (soit plus de 20 %) selon le compilateur

⇒ bornes d'erreur d'évaluation prouvées avec Gappa

→ borne d'erreur :  $2^{-60}$  à  $2^{-69}$

	CRlibm	
	Horner	KE/Estrin
gcc-3.4	342 cycles	259 cycles
gcc-4.0	342 cycles	269 cycles

## Exemple 2 – Accélération du logarithme népérien

### Logarithme existant

- ▶ évaluation d'un polynôme de degré 7 sur  $[-2^{-7}, 2^{-7}]$   
→ évalué par un schéma de Estrin

### Solution proposée

- ▶ schéma hybride Paterson & Stockmeyer / Estrin
- ⇒ gain : 3 à 5 cycles selon le compilateur
- ⇒ borne d'erreur d'évaluation prouvée avec Gappa
- même borne d'erreur qu'avec Estrin :  $2^{-62}$

	CRLibm	
	Estrin	PS/Estrin
gcc-3.4	143 cycles	138 cycles
gcc-4.0	151 cycles	148 cycles

- ⇒ **intégré à CRLibm**

## Conclusions et travaux en cours

### Conclusions

- ▶ analyse de l'algorithme de Knuth & Eve
  - preuve de stabilité *backward*, nombre de conditionnement, expérimentations
- ▶ propositions de schémas hybrides : bon compromis précision / rapidité
- ▶ application aux fonctions élémentaires
  - accélération de la fonction  $\ln(x)$  de CRlibm
- ▶ preuve des résultats avec Gappa

### Travaux en cours

- ▶ détermination d'une meilleure combinaison « décalage/permutation » pour l'algorithme de Knuth & Eve
  - ▶ automatisation de la conception des schémas d'évaluation hybrides
- ⇒ **accélération de la phase de préparation**